



TIME

Arduino provides four different time manipulation functions:

- delay() function
- delayMicroseconds() function
- millis() function
- micros() function

## delay() function

delay() function accepts a single integer (or number) argument. This number represents the time (measured in milliseconds). The program should wait until moving on the next line of code when it encounters this function.

*delay() is not a good way to make your program wait, because it's known as a "blocking" function.*

## EXAMPLE flashing LED

```

/* Flashing LED
 * -----
 * Turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 2 seconds. *
 */

int ledPin = 13; // LED connected to digital pin 13

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() {
  digitalWrite(ledPin, HIGH); // sets the LED on
  delay(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delay(1000); // waits for a second
}

```

## delayMicroseconds() function

delayMicroseconds() function accepts a single integer (or number) argument. This number represents the time and is measured in microseconds. There are thousand microseconds in millisecond, and a million microseconds in a second.

Currently, the largest value that can produce an accurate delay is 16383. For delays longer than a few thousand microseconds, you should use the delay() function instead.

## EXAMPLE flashing LED

```
/* Flashing LED
 * -----
 * Turns on and off a light emitting diode(LED) connected to a digital
 * pin, in intervals of 1 seconds. *
 */

int ledPin = 13; // LED connected to digital pin 13

void setup() {
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop() {
  digitalWrite(ledPin, HIGH); // sets the LED on
  delayMicroseconds(1000); // waits for a second
  digitalWrite(ledPin, LOW); // sets the LED off
  delayMicroseconds(1000); // waits for a second
}
```

## millis() function

This function is used to return the number of milliseconds at the time, the Arduino board begins running the current program. This number overflows i.e. goes back to zero after approximately 50 days.

*This function returns milliseconds from the start of the program.*

### EXAMPLE

```
■ unsigned long time; void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    Serial.print("Time:"); time = millis();  
    //prints time since program started  
    Serial.println(time);  
    // wait a second so as not to send massive amounts of data  
    delay(1000);  
}
```

## micros() function

The `micros()` function returns the number of microseconds from the time, the Arduino board begins running the current program. This number overflows i.e. goes back to zero after approximately 70 minutes. On 16 MHz Arduino boards (e.g. Duemilanove and Nano), this function has a resolution of four microseconds (i.e. the value returned is always a multiple of four). On 8 MHz Arduino boards (e.g. the LilyPad), this function has a resolution of eight microseconds.

## EXAMPLE

```
unsigned long time; void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.print("Time:");  
  time = micros(); //prints time since program started  
  Serial.println(time); // wait a second so as not to send massive amounts of data  
  delay(1000);  
}
```